

Implementação remota de uma *dashboard* interativa para exibição de métricas simultâneas dos *host* de um *cluster* do tipo *Beowulf*

Eduardo de Almeida

Dezembro, 2020

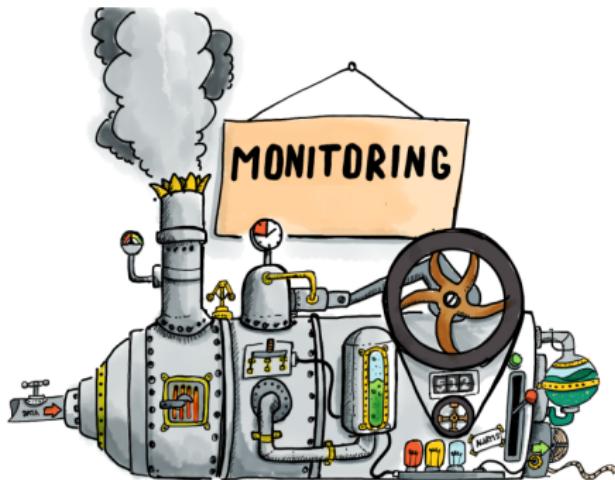
Apresentação do problema

- Dificuldade de monitoramento da saúde do sistema *Clustering*.
- Necessidade de depuramento de erros.
- Restrição geográfica para *troubleshooting*.
- Falta de parâmetros quantitativos para otimização.



Por quê monitorar métricas?

- Prevenir sobrecarga.
- Evitar danos permanente *hardware*.
- Possibilita tomada rápida de decisão.
- Permite *Troubleshooting*.
- Fornece informes quantitativos atualizados sobre a saúde e eficiência do sistema.
- Evita desperdício de recursos computacionais ao se buscar implementação mais otimizada.



Por quê *Netdata*

- *Open Source.*
- *GitHub.*
- Integração com back-end e frameworks.



Por quê Netdata

- *Dashboard interativa.*
- Customização.
- Armazenamento.
- *Streaming.*
- Escabilidade e baixo uso de Recursos computacionais.
- Integração.



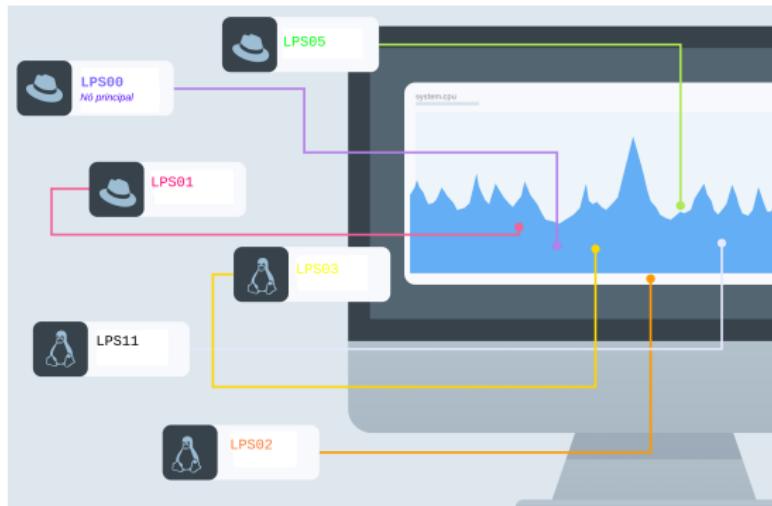
Motivação

- Monitoramento em tempo real de métricas.
- TTD e TTM.
- Alarmes e notificações.
- Solução Remota e segura.



Objetivo

- Exibir informações simultâneas sobre quaisquer métricas em todos os nós do sistema de *Clustering*.



Objetivo

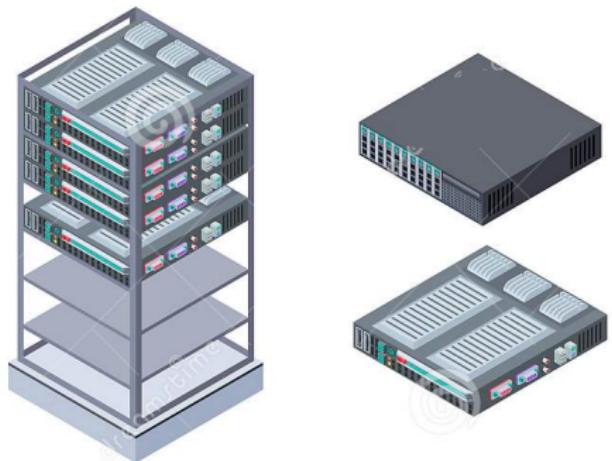
- Acessar remotamente em qualquer *browser* de qualquer dispositivo os métricas individuais de cada nó do sistema de *Clustering*.



- Estrutura do sistema de *Cluster*.
- Sistema operacional.
- Mapeamento da rede e identificação dos *host*.
- Instalação e configuração do *software*.
- Implementação da ferramenta de *Streaming*.
- Comunicação *API*.
- *Multi-Dashboard*.
- Implementação Remota.

Estrutura do Sistema

- *Cluster Beowulf.*
- Nós conectados por *Switch FastEthernet*.
- *front-end* com 2 *NIC* e acesso á internet por roteador.
- *PVI* e *MPI*.
- Protocolo pilha *TCP/IP*.
- Servidor *SSH*.

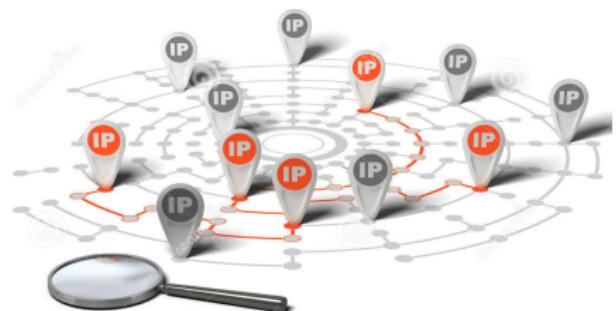


Sistema Operacional

- *Linux.*
- Distribuição *Fedora* e *Ubuntu*.
- Terminal *CLI, BASH*.
- *FHS*.
- Repositório.
- *OpenSSH*.
- Comandos.



- Pilha de protocolos *TCPIP*.
- Endereçamento *IPv4*.
- Endereçamento *MAC*.
- Endereçamento de Portas.



Mapeamento da Rede

- *IPv4*
- *estático.*
- *IFCONFIG.*
- *Route.*
- *NMAP.*
- *IP de loopback*

```
Last login: Thu Nov 12 16:25:24 2020 from 201.21.142.228
[...]
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.1 brd 255.255.255.0 broadcast 192.168.1.255
        netmask 255.255.255.0
        ether 00:09:95:68:9c:fe txqueuelen 1000 (Ethernet)
        RX packets 302144444 bytes 266216463923 (247.9 GiB)
        RX errors 0 dropped 24 overruns 0 frame 0
        TX packets 185047852 bytes 1601269839 (14.9 GiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
        device interrupt 20 memory 0xf7400000-0xf7420000

enp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 19  brd 255.255.255.0 broadcast 5.255
        netmask 255.255.255.0
        ether 00:0e:05:31:14:02 txqueuelen 1000 (Ethernet)
        RX packets 203180073 bytes 14895931436 (13.8 GiB)
        RX errors 0 dropped 143898 overruns 0 frame 0
        TX packets 7941471 bytes 2308221783 (2.1 GiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 brd 255.0.0.0
        netmask 255.0.0.0
        ether 00:00:00:00:00:00 txqueuelen 1000 (Loopback Local)
        RX packets 194311 bytes 114329727 (109.0 MiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 194311 bytes 114329727 (109.0 MiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
Host ls up (0.00094s latency).
Host shown: 192.168.1.1 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind
2049/tcp  open  nfs
9090/tcp  open  zeus-admin
19999/tcp open  dnpsec

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.86 seconds
```

Instalação

- Acesso *SSH*.
- Pacotes de dependências.
- *Curl*.
- *Git*.
- *Firewall*.
- Inicialização.
- Instalador.

```
1 # Download do package em FEDORA
2
3 # Atualizar dependencias do sistema
4 sudo yum update
5
6 # Instalar comando curl
7 sudo yum install curl
8
9 # Dependencias adicionais Git
10 sudo yum install asciidoc xsltproc docbook2X getopt
11
12 # Download do conteúdo da url para localização especificada
13 sudo curl -s "https://raw.githubusercontent.com/netdata/netdata/master/packaging/installer/-install-required-packages.sh" >/tmp/install-required-packages.sh && bash /tmp/install-required-packages.sh -l netdata-all
14
15 # Instalar pacotes baixados
16 sudo dnf install zlib-devel libuv-devel libuv-devel lz4-devel Judy-devel openssl-devel
elfutils-libelf-devel libmnl-devel gcc make git autoconf autoconf-archive autonake
pkgconfig curl findutils python cmake
17
18 # Instalar comando Git
19 sudo yum install git-all
20
21 # Descarregar repositório - pasta 'netdata' é criado em /home/lps
22 git clone https://github.com/netdata/netdata.git --depth=100
23 cd netdata
24
25 # Executar script em /home/lps/netdata/tanetdata-installer.sh
26 sudo ./netdata-installer.sh
27
28 #Iniciar aplicação no boot
29 sudo systemctl enable netdata
30
31 #Adicionar exceção Firewall
32 sudo firewall - cmd --add-port=19999/tcp
33 firewall -cmd --runtime-to-permanent
34 firewall -cmd --reload
```

Dashboard principal

- A partir dessa etapa é possível acessar cada dashboard individualmente.
- Por meio do endereço de IP Local de cada nó
http://X.Y.Z.U:19999 individualmente.
- Todos nós podem ser acessados remotamente por *SSH Tunneling*.
- Definidos parâmetros de métricas, classes de gráficos no Diretório
usr/share/netdata/web/gui.

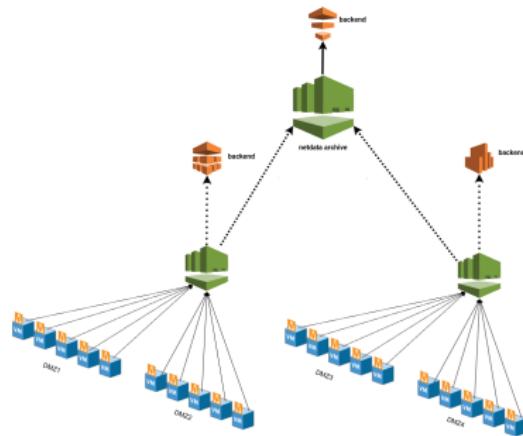


- SSH atribui diretório *home/lps* automaticamente.
- Retroceder diretório raiz.
- etc/netdata Para configurações.
- usr/share/netdata/web para HTML.
- Código JavaScript customizável localizado em *usr/share/netdata/web/dashboard_info.js*

```
^ _.-_ _.-_ _.-_ _.-_ _.-_ _.-_ netdata  
| real-time performance monitoring, done right!  
+-----+-----+-----+-----+-----+-----+-----+----->  
  
You are about to build and install netdata to your system.  
  
The build process will use /tmp for  
any temporary files. You can override this by setting $TMPDIR to a  
writable directory where you can execute files.  
  
It will be installed at these locations:  
  
- the daemon      at /usr/sbin/netdata  
- config files   in /etc/netdata  
- web files     in /usr/share/netdata  
- plugins        in /usr/libexec/netdata  
- cache files   in /var/cache/netdata  
- db files       in /var/lib/netdata  
- log files      in /var/log/netdata  
- pid file       at /var/run/netdata.pid  
- logrotate file at /etc/logrotate.d/netdata  
  
This installer allows you to change the installation path.  
Press Control-C and run the same command with --help for help.
```

Streaming

- Utiliza-se parâmetros de rede.
- IP Estático + API.
- MGUID e UUIDs.
- DB em localhost.
- etc/netdata para configurações.
- SUDO VIM.



Topologias

target	memory mode	web mode	stream enabled	backend	alarms	dashboard
headless collector	none	none	yes	only for <code>data source = as collected</code>	not possible	no
headless proxy	none	not none	yes	only for <code>data source = as collected</code>	not possible	no
proxy with db	not none	not none	yes	possible	possible	yes
central netdata	not none	not none	no	possible	possible	yes

Nó Principal

- *etc/netdata* para configurações.
- *SUDO VIM*.
- Habilitar modo memória e web.
- *Stream Disable*.
- Especificar *API Autenticação e Identificação*
- Especificar *MGUID* se necessário roteamento métricas.

```
# -----
# 2. ON PARENT NETDATA - THE ONE THAT WILL BE RECEIVING METRICS
#
# You can have one API key per child,
# or the same API key for all child nodes.
#
# netdata searches for options in this order:
#
# a) parent netdata settings (netdata.conf)
# b) [stream] section      (above)
# c) [API_KEY] section     (below, settings for the API key)
# d) [MACHINE_GUID] section (below, settings for each machine)
#
# You can combine the above (the more specific setting will be used).
#
# API key authentication
# If the key is not listed here, it will not be able to push metrics.
#
# [API_KEY] is [YOUR-API-KEY], i.e. [11111111-2222-3333-4444-555555555555]
# [2xxxxx-xxxxx-11ex-aex-xxxxxx] # Default settings for this API key
#   # Default settings for this API key
#
# You can disable the API key, by setting this to: no
# The default (for unknown API keys) is: no
enabled = yes
#
# A list of simple patterns matching the IPs of the servers that
# will be pushing metrics using this API key.
# The metrics are received via the API port, so the same IPs
# should also be matched at netdata.conf [web].allow connections from
allow_from = *
#
# The default history in entries, for all hosts using this API key.
# You can also set it per host below.
# If you don't set it here, the history size of the central netdata
# will be used.
default_history = 3600
#
# The default memory mode to be used for all hosts using this API key.
# You can also set it per host below.
# If you don't set it here, the memory mode of netdata.conf will be used.
# Valid modes:
-- INSERÇÃO --
```

Nós secundários

- *etc/netdata* para configurações.
- *SUDO VIM*.
- Habilitar modo memória e web.
- *Stream Enable*.
- Especificar *API Autenticação e Identificação*
- Especificar *HOST* por meio *IPv4 Estático* e porta.

```
# netdata configuration for aggregating data from remote hosts
#
# API keys authorize a pair of sending-receiving netdata servers.
# Once their communication is authorized, they can exchange metrics for any
# number of hosts.
#
# You can generate API keys, with the linux command: uuidgen
#
# -----
# 1. ON CHILD NETDATA - THE ONE THAT WILL BE SENDING METRICS
#
[stream]
    # Enable this on child nodes, to have them send metrics.
    enabled = yes

    # Where is the receiving netdata?
    # A space separated list of:
    #
    # [PROTOCOL:]HOST[%INTERFACE][[:PORT]][:SSL]
    #
    # If many are given, the first available will get the metrics.
    #
    # PROTOCOL = tcp, udp, or unix (only tcp and unix are supported by parent nodes)
    # HOST     = an IPv4, IPv6 IP, or a hostname, or a unix domain socket path.
    #           IPv6 IPs should be given with brackets [ip:address]
    # INTERFACE = the network interface to use (only for IPv6)
    # PORT     = the port number or service name (/etc/services)
    # SSL      = when this word appear at the end of the destination string
    #           the Netdata will encrypt the connection with the parent.
    #
    # This communication is not HTTP (it cannot be proxied by web proxies).
    destination = A.B.C.D:19999
```

Script para alterar parâmetros Streaming

- *pathname* absolutos.
- *CAT*.
- *EOF*.
- *>*.
- *«*.

```
1 #Configuracao Stream
2
3 sudo cat >/etc/netdata/netdata.conf <<EOF
4 [global]
5   memory mode = dbengine
6 [web]
7   mode = static-threaded
8 EOF
9
10 sudo cat >/etc/netdata/stream.conf <<EOF
11 [stream]
12   enabled = yes
13   destination = A.B.C.D:19999
14   api key = API KEY GERADA
15 EOF
--
```

Visualização comunicação API por *Streaming*

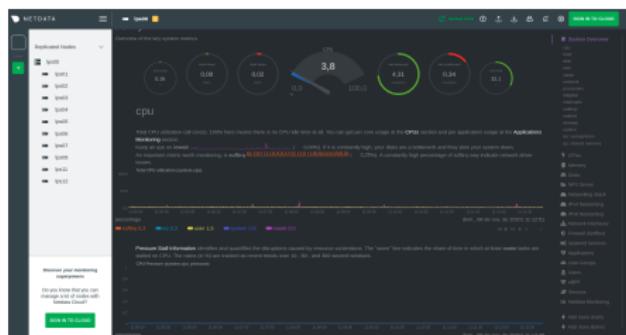
```
{  
    "version": "v1.26.0-138-nightly",  
    "uid": "dc8a  
    "mirrored_hosts": [  
        "lps00",  
        "lps00",  
        "lps11",  
        "lps09",  
        "lps09",  
        "lps05",  
        "lps05",  
        "lps02",  
        "lps01",  
        "lps04"  
    ],  
    "mirrored_hosts_status": [  
        {"guid": "d1", "i-2068-", -806, "i0b2", "reachable": true, "claim_id": "dc8af2"},  
        {"guid": "e0", "i-1fa5b-", -844, "i0fd", "reachable": true, "claim_id": null},  
        {"guid": "b1", "i-2076-", -346, "i93c", "reachable": true, "claim_id": null},  
        {"guid": "4f", "i-2075-", -237, "i421", "reachable": true, "claim_id": null},  
        {"guid": "d5", "i-2074-", -346, "i004", "reachable": true, "claim_id": null},  
        {"guid": "9f", "i-2071-", -ed5, "i102", "reachable": true, "claim_id": null},  
        {"guid": "51", "i-2070-", -c51, "i35f", "reachable": true, "claim_id": null},  
        {"guid": "7f", "i-206f-", -391, "i8be", "reachable": true, "claim_id": null},  
        {"guid": "2f", "i-206d-", -846, "i3fc", "reachable": true, "claim_id": "33fbdc"},  
        {"guid": "8b", "i-2064-", -846, "i89d", "reachable": true, "claim_id": "33fbdc"},  
        {"guid": "80", "i-206a-", -401, "i839", "reachable": true, "claim_id": null},  
        {"guid": "11", "i-206f-", -832, "i726", "reachable": true, "claim_id": "175e00"}  
    "alarms": {  
        "email": 0,  
        "warning": 1,  
        "critical": 0  
    },  
    "os_name": "Fedora",  
    "os_id": "fedora",  
    "os_label": "LinuxUnknown",  
    "os_version": "31 (Server Edition)",  
    "os_version_id": "31",  
    "os_detection": "/etc/os-release",  
    "cores_total": "4",  
    "total_disk_size": "48000797868032",  
    "cpu_freq": "3700000000",  
    "ram_total": "8319168512",  
    "container_os_name": "none",  
    "container_os_id": "none",  
    "container_os_like": "none",  
    "container_os_version": "none",  
    "container_os_version_id": "none",  
    "kernel_name": "Linux",  
    "kernel_version": "5.4.6_64",  
    "architecture": "x86_64",  
    "virtualization": "none",  
    "virt_detection": "qemu-detect-virt",  
    "container": "unknown",  
    "container_detection": "none",  
},  
}
```

<http://A.B.C.D:19999/api/v1/info>

- JSON será parâmetro para implementação da *Multi-Dashboard*.
- Mirrored Host Status deve estar Recheable.

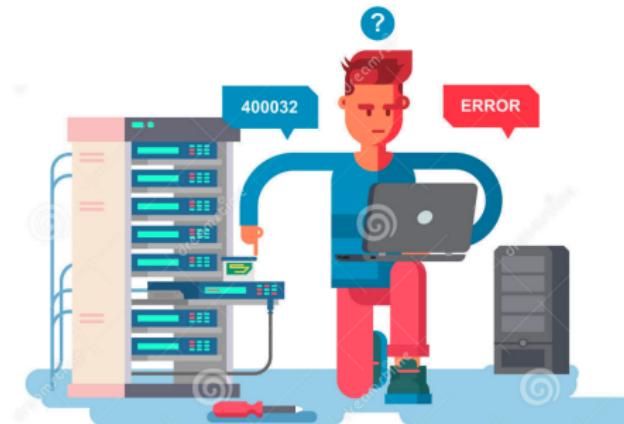
Resultado *Streaming*

- A partir dessa etapa todos nós estão disponíveis no menu superior esquerdo da *dashboard* do nó principal.
- Navegação imediata.
- Acesso Remoto á todos os nós.
- <http://X.Y.Z.W:19999> ou por *SSH Tunneling*.
- Sendo X.Y.Z.W IP real do *front-end*.



Resultado *Streaming*

- Depuração imediata.
- Intervenção por SSH.
- Acesso á DB de métricas coletadas para mapear causas de problemas e identificar erros.
- Controle de Granularidade.
- Gerenciamento do Cluster remotamente.



JS e HTML

- Diretório
usr/share/netdata/web.
- dashboard_info.js
- dash-example.html.
- sudo cp
/usr/share/netdata/web/dash-example.html
/usr/share/netdata/web/lps.html,
- .
- sudo vim lps.html.
- Customizar classe, tipo de dado, dimensão. (definidas em /usr/share/netdata/web/gui)

```
1 <div class="dash-graph" # Usar a classe dash-graph
2 data-dash-netdata="system.cpu" # Usar data-dash-netdata para definir tipo dado
3 data-dygraph-valuerange="[0, 100]"> # Adicional: inclusao de qualquer outra data-* # Usar a classe dash-graph
# Usar data-dash-netdata para definir tipo dado
# Dimensao que sobrescreve definicao padrao
# Titulo que sobrescreve definicao padrao
# Unidade que sobrescreve definicao padrao
4 </div>
5 <div class="dash-chart" # Usar dash-chart para graficos tipo pie chart
6 data-dash-netdata="system.io" # Usar data-dash-netdata para definir tipo dado
7 data-dimensions="ln" # Dimensao que sobrescreve definicao padrao
8 data-title="Disk Read" # Titulo que sobrescreve definicao padrao
9 data-common-units="dash.lo"> # Unidade que sobrescreve definicao padrao
10 </div>
```

- Classes definidas podem ser aplicadas sob um contêiner genérico html de modo a trata-ló globalmente.
- Explicitar o IP estático e porta para o front-end em var dash = new Dash('http://A.B.C.D:19999');
- Sintaxe dash-* utiliza como parâmetro no arquivo html a JSON <div class="netdatahoststatscontainer template">

Resultado *Multi-Dashboard*

- Nós da esquerda para direita.
- Exibição em um único *display*.
- Acesso remoto por meio de *SSH Tunneling*.



Multi-Dashboard

